

# APLIKASI PENULISAN NOTASI BALOK DARI *FILE* MIDI *MONOPHONIC*

**Liliana, Sukanto Tedjokusuma, Richie Alamveta**

Jurusan Informatika, Fakultas Teknologi Industri, Universitas Kristen Petra

Jl. Siwalankerto 121-131, Surabaya, 60236

E-mail: lilian@petra.ac.id, sukanto@petra.ac.id

**ABSTRAK:** Musik digemari banyak kalangan. Tidak hanya untuk dinyanyikan tetapi juga untuk dimainkan. Kesulitan yang sering terjadi adalah file musik yang didapatkan tidak disertai dengan partitur musiknya. Hal ini menyebabkan *user* kesulitan untuk memainkan lagu tersebut. Aplikasi ini dibuat dengan tujuan untuk membantu *user* dapat membuat partitur sendiri dari *file* midi. Aplikasi ini dibuat untuk menghasilkan partitur secara otomatis dalam bentuk notasi balok. Aplikasi ini dibuat dengan menggunakan Borland Delphi 7. *Input* untuk perangkat lunak ini berupa *file* midi, di mana *file* midi tersebut harus terdiri dari satu instrumen saja. Perangkat lunak akan membaca dan memproses *file* midi menjadi data not, kemudian data not disimpan dalam struktur data. *Output* dari perangkat lunak ini berupa gambar penulisan notasi balok dari *file* midi yang dipilih *user*. Partitur dari aplikasi ini nantinya dapat disimpan dan dicetak. Aplikasi ini dapat menggambarkan notasi balok sesuai dengan data yang terdapat dalam *file* midi, tetapi masih terdapat keterbatasan, misalnya penggambaran simbol tanda henti dalam jumlah yang banyak selalu dimulai dengan nilai yang terbesar..

**Kata kunci:** *File* midi, notasi balok, partitur, monophonic.

**ABSTRACT:** Music is interested for many people, from different generations. Not only to be sung but also to be played. The most matter is midi music files downloaded from any website are not combined with their music sheet. This condition makes users have a difficulty in playing the song. Consider this problem, this application is developed to support user with partitur crated from midi music file. This application is designed to produce a music sheet aotomatically in music notation. This application is created using Borland Delphi 7. This application receives monophonic midi music file as input. This application will read and processes the midi file be a music string, then the music string will be kept in a data structure. Output from this application is a music sheet from midi music file chosen by user. User also can save and print the music sheet. This application success to create the music sheet from midi music file, but it still has limitation, for example is drawing the rest symbol. If there are so many rest symbols, application will draw from the rest symbol whose biggest value.

**Keywords:** *Midi* music file, music notation, music sheet, monophonic.

## PENDAHULUAN

Ketersediaan partitur musik dalam bentuk file cukup terbatas, tidak sebanding dengan ketersediaan file musik, khususnya dalam format midi. Sementara itu, kebutuhan dari para pemusik dalam melakukan improvisasi musik atau aransemen ulang juga memerlukan kemudahan teknologi ini. Oleh karena itu, para peneliti dapat mengembangkan aplikasi yang dapat menuliskan kembali file musik dalam bentuk notasi balok yang sumbernya dari *file* midi. Pemilihan *file* midi karena *file* jenis ini adalah yang paling umum digunakan dan sudah menyimpan data not-not yang dimainkan. Not balok dipilih karena merupakan notasi yang umum digunakan oleh seorang pemusik. Dengan aplikasi tersebut, maka *user* dapat membuat sendiri *sheet music* dari *file* midi.

*File* midi sendiri dapat dengan mudah diunduh user lewat internet sehingga user dapat mencari lagu mana yang ingin dipelajarinya. Ukuran dari *file* midi relatif kecil, hanya beberapa *KiloBytes*, sehingga user dapat mengunduhnya dengan cepat. Selain memanfaatkan internet, *file* midi juga dapat dibuat sendiri dengan keyboard musik. Pemain musik yang menyimpan lagu dari keyboard musik umumnya berupa *file* midi juga.

Kesulitan yang mendapat perhatian besar dalam pembuatan aplikasi ini adalah mendeteksi durasi sebuah not dan durasi sebuah tanda istirahat. Selain itu, dalam penggambaran notasi balok juga terdapat kesulitan dalam menggambar not berdampingan yang mempunyai durasi yang berbeda.

Dalam pengembangannya, aplikasi ini dapat dijadikan pemutar *file* midi atau dapat juga ditambah-

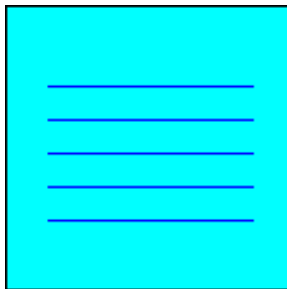
kan gambar tuts piano yang dapat memberi informasi kepada *user* bagaimana cara bermainnya, sehingga *user* dapat mengetahui kapan harus menekan not dan not mana yang harus ditekan. Pengembangan yang lain, aplikasi ini dapat juga dijadikan *sheet music editor*.

## TEORI MENGENAI NOT BALOK

Dalam notasi musik, nada dilambangkan oleh not. Setiap nada memiliki frekuensi yang berbeda. Masing-masing nada digambarkan pada posisi yang berbeda pula. Notasi balok adalah standar yang digunakan dalam penulisan notasi musik. Notasi balok didasarkan pada paranada dengan lambang untuk tiap nada menunjukkan durasi dan ketinggian nada tersebut. Tinggi nada digambarkan secara vertikal sedangkan waktu/ritme digambarkan secara horizontal. Durasi nada ditunjukkan dalam ketukan [1, 2].

- **Paranada (*Musical Staff*)**

Garis Paranada menggambarkan urutan waktu dari kejadian musik. Paranada dibentuk oleh 5 buah garis yang dipisahkan oleh 4 buah spasi, seperti terlihat pada Gambar 1.



**Gambar 1. Paranada (*Musical Staff*)**

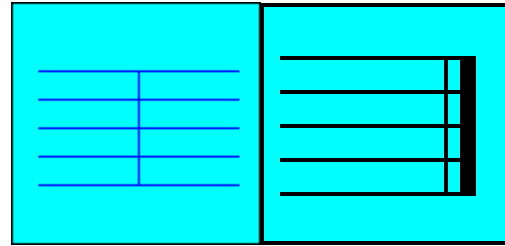
Not yang terletak di garis atau spasi lebih tinggi berarti memiliki tinggi nada yang lebih tinggi. Not pada paranada dibaca dari kiri ke kanan. Apabila penulisan membutuhkan posisi not yang melebihi paranada, dapat menggunakan garis tambahan.

- **Garis Bar (*Bar lines*)**

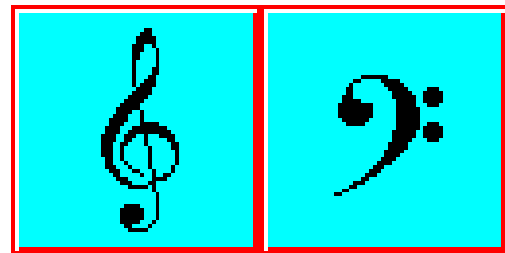
Garis Bar diletakkan pada paranada untuk memisahkan kumpulan ketukan sesuai dengan tanda birama. Jarak antara garis bar disebut dengan *measure* atau bar. Terdapat pula tanda garis bar ganda, yang menandakan bahwa akhir dari suatu notasi sudah tercapai. Garis bar dan bar ganda dapat dilihat pada Gambar 2.

- **Tanda Kunci (*Clef Symbol*)**

Tanda kunci menunjukkan rentang not dari karya musik yang dimainkan. Tanda kunci yang umum digunakan adalah tanda kunci *treble* dan tanda kunci *bass*, seperti pada Gambar 3.



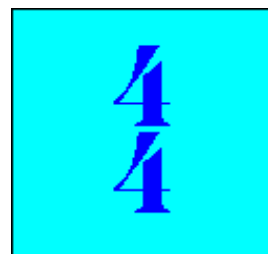
**Gambar 2. Garis Bar/*Bar Lines* (kiri), garis bar ganda/*Double Bar Line* (kanan)**



**Gambar 3. Tanda kunci G/*Treble* (kiri), tanda kunci F/*Bass* (kanan)**

- **Tanda Birama (*Time Signature*)**

Angka di atas pada tanda birama menunjukkan banyaknya ketukan pada sebuah bar, sedangkan angka di bawah pada tanda birama menunjukkan nilai not yang dianggap sebagai satu ketuk. Tanda birama diletakkan setelah tanda kunci dengan tulisan seperti pada Gambar 4.



**Gambar 4. Tanda Birama (*Time Signature*)**

Terdapat bermacam-macam tanda birama, seperti *simple duple* (2/4), *simple triple* (3/4), *simple quadruple* (4/4), dan lain sebagainya.

- **Not (*Note*)**







Not adalah representasi dari nada di dalam notasi musik. Not memiliki informasi berupa tinggi nada dan durasinya. Gambar 5 menunjukkan posisi dari tiap-tiap not dalam tanda kunci *treble* dan *bass*.



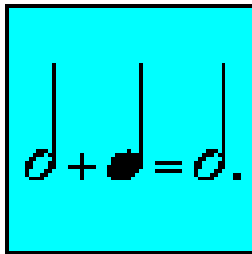
**Gambar 5. Posisi Not dalam Tanda Kunci *Treble* dan *Bass***

Untuk durasi dari suatu not digambarkan dengan simbol not yang berbeda, seperti terdapat pada Tabel 1.

**Tabel 1. Jenis-jenis Nilai Not**

Not	Nilai	Not	Nilai
	Not Penuh (1)		Not Setengah (1/2)
	Not Seperempat (1/4)		Not Seperdelapan (1/8)
	Not Seperenambelas (1/16)		Not Sepertigadua (1/32)







- **Not Bertitik (*Dotted Note*)**  
Titik diletakkan di sebelah kanan sebuah not, berarti durasi not tersebut bertambah setengah dari durasi aslinya, seperti tampak pada Gambar 6.



**Gambar 6. Not Bertitik (*Dotted Note*)**

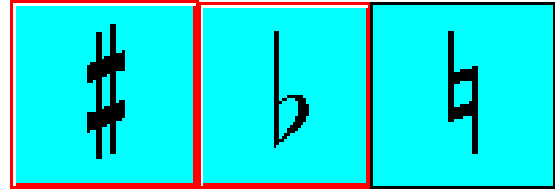
- **Tanda Henti (*Rest*)**  
Tanda henti menunjukkan tidak adanya nada pada saat waktu tersebut. Tanda henti juga memiliki durasi. Namun tanda henti tidak memiliki informasi tinggi nada. Tabel 2 menunjukkan simbol-simbol yang menggambarkan nilai dari suatu tanda henti.

**Table 2. Jenis-jenis Nilai Tanda Henti**

Henti	Nilai	Henti	Nilai
	Istirahat Penuh (1)		Istirahat Setengah (1/2)
	Istirahat Seperempat (1/4)		Istirahat Seperdelapan (1/8)
	Istirahat Seperenambelas (1/16)		Istirahat Sepertigadua (1/32)

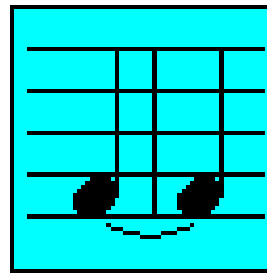
- ***Accidental***  
*Accidental* adalah simbol yang digunakan untuk menaikkan atau menurunkan tinggi nada dari suatu not. Simbol ini diletakkan di sebelah kiri suatu not. Terdapat 3 simbol *accidental*, yaitu *sharp* (menaikkan nada sebesar 1 *semitone*), *flat* (menurunkan nada sebesar 1 *semitone*), dan *natural* (membatalkan semua *accidental* pada nada tersebut). Sebagai contoh, untuk menghasilkan not

yang menunjukkan nada C#, dapat digunakan simbol *sharp* sebelum nada C. Simbol *accidental* yang lengkap terlihat pada Gambar 7.



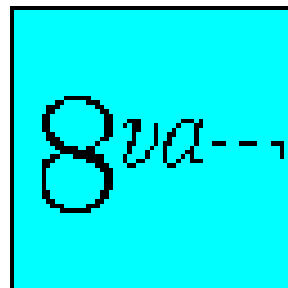
**Gambar 7. Tanda Kres/Sharp (kiri), Tanda Mol/Flat (tengah), Tanda Pugar/Natural (kanan)**

- ***Tie***  
*Tie* menghubungkan nilai suatu not dengan not yang lain, lihat Gambar 8. Biasanya digunakan untuk not yang nilainya melebihi nilai dari suatu bar.



**Gambar 8. Tanda *Tie***

- ***Octava***  
*Octava* menunjukkan not-not pada paranada yang diberi garis dengan tanda ini dimainkan satu oktaf lebih tinggi dari nilai aslinya. Tanda *octava* diletakkan di atas paranada, seperti terlihat pada gambar 9. *Octava bassa* menunjukkan bahwa not-not pada paranada yang diberi garis dengan tanda ini dimainkan satu oktaf lebih rendah. Tanda *octava bassa* diletakkan di bawah paranada.

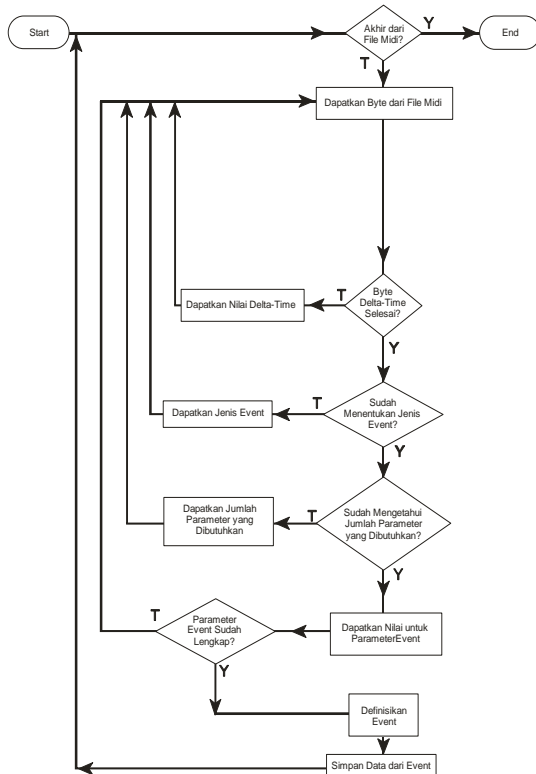


**Gambar 9. Octava**

## DESAIN APLIKASI

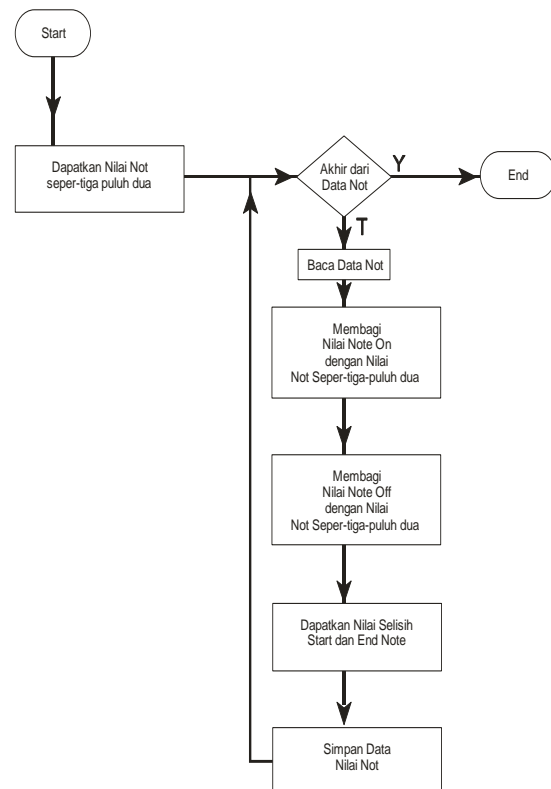
Pada bagian ini dibahas mengenai desain sistem dari aplikasi penulisan notasi balok ini. Secara garis besar, sistem aplikasi penulisan notasi balok ini terdiri dari empat bagian [3, 4, 5], yaitu:

- Pembacaan *file* midi  
Hasil pembacaan *file* midi masih berupa nilai-nilai dalam bentuk desimal. Nilai-nilai ini kemudian dikonversikan ke dalam bentuk *byte* hexadecimal agar dapat didefinisikan sesuai dengan teori *file* midi.
- Mendefinisikan isi dari *file* midi  
Pertama, aplikasi mendefinisikan header dari file midi, kemudian track header dan track event. Pendefinisian tersebut akan menghasilkan data yang diperlukan, diantaranya tipe file midi, time division, dan track event. Track event mengandung data musik seperti tempo dari lagu, birama, kapan not harus dimainkan dan berhenti, not apa yang dimainkan, dan lain-lain.  
Dalam mendefinisikan isi file midi, *time division* merupakan data yang penting. *Time division* merupakan sebuah nilai yang digunakan sebagai acuan untuk menentukan nilai suatu nada. Nilai yang terdapat pada time division sama dengan nilai delta-time untuk not seperempat. Data lain yang dapat terdefinisi pada bagian ini, yaitu nilai birama. Birama digunakan untuk mengetahui jumlah nilai not yang dapat diletakkan pada suatu bar. Data ini nantinya dapat digunakan untuk mendapatkan posisi not terletak pada bar ke berapa. Apabila perangkat lunak tidak mendeteksi event birama ini, maka birama dianggap bernilai 4/4. Flowchart prosedur pendefinisian isi file midi dapat dilihat pada Gambar 10.



**Gambar 10. Flowchart Prosedur Pendefinisian Isi File Midi**

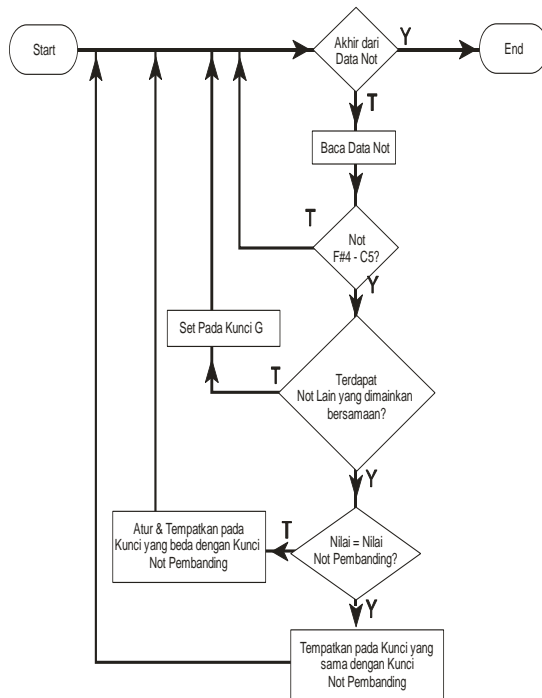
- Pengolahan tiap-tiap not  
Pada proses ini terdapat beberapa sub proses, yaitu:
  - a. Desain Penghitungan Nilai Not  
Proses ini dilakukan dengan membagi nilai status *on* dan *off* dengan nilai not seper-tiga puluh dua. Hasil pembagian ini harus selalu dibulatkan agar nilai not merupakan kelipatan nilai seper tiga puluh dua. Pembagian dengan nilai not seper-tiga puluh dua karena pada perangkat lunak ini nilai not terkecil yang digunakan yaitu not seper tiga puluh dua, sedangkan nilai *time division* adalah nilai not seperempat. Oleh karena itu, untuk mendapatkan nilai not seper-tiga puluh dua dilakukan dengan membagi nilai *time division* dengan 8. Setelah mendapatkan nilai *on* dan *off* yang baru, maka proses selanjutnya adalah mencari selisihnya dengan cara mengurangkan kedua nilai tersebut. Gambar 11 memperlihatkan flowchart penghitungan nilai not secara lengkap.



**Gambar 11. Flowchart Prosedur Penghitungan Nilai Suatu Not**

- b. Desain Penentuan Posisi Suatu Not  
Proses ini dilakukan dengan menghitung banyaknya ketukan not  $\frac{1}{32}$  dalam satu bar. Banyak ketukan  $\frac{1}{32}$  dalam 1 bar = (Numerator \* 32)/Denominator.
- c. Desain Penempatan Not Pada Kunci *Treble* atau *Bass*.

Proses penempatan not pada kunci treble atau bass dituliskan secara lengkap seperti pada Gambar 12.



**Gambar 12. Flowchart Prosedur Penempatan Not Pada Kunci Treble Atau Bass**

d. Desain Pemberian Status Pada Not Berbendera Sambung

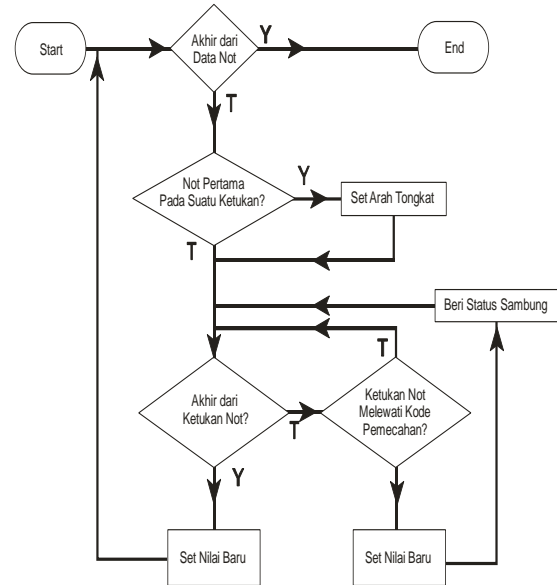
Pada penggambaran not balok, jika terdapat beberapa not berbendera, maka penggambaran not dapat diganti dengan not berbendera sambung. Pada aplikasi yang dikembangkan ini, dilakukan pengecekan apakah nada selanjutnya bernilai sama. Biasanya not berbendera sambung berjumlah 2 atau 4. Apabila terdapat 3 not berbendera sambung, maka ada kemungkinan untuk menyambung dengan not berikutnya, tetapi nilai not berikutnya harus bernilai lebih besar dari nilai not berbendera. Dalam kasus tersebut, not akan disambungkan, tetapi not tersebut akan dipecah menjadi 2 not sesuai dengan nilai aslinya.

e. Desain Pemberian Kode Pemecahan Not

Jika terdapat not dengan nilai tertentu yang tidak dapat digambarkan dengan satu not saja, not harus dipecah sehingga penggambarannya nanti akan menjadi 2 not atau lebih. Penggambaran not akan ditambahkan dengan simbol *tie*.

f. Desain Penyimpanan Data Not Baru

Proses penyimpanan data not baru dapat dilihat pada flowchart di Gambar 13.



**Gambar 13. Flowchart Penyimpanan Data Not Baru**

• Penulisan notasi balok

Setiap not yang tersimpan memiliki data berupa simbol not tersebut, posisi not, simbol-simbol lain yang merupakan atribut dari not. Simbol not yang digambarkan tergantung pada nilai dari not yang terbaca. Peletakan simbol not pada gambar paranada ditentukan oleh posisi not dan setelah itu menggambar atribut not lainnya. Pada proses penggambaran ini, jika jumlah not pada suatu ketukan sama dengan 0, maka akan diberi tanda henti. Selain menggambar not dan tanda henti, pada bagian ini aplikasi juga melakukan penggambaran simbol-simbol lain seperti tanda *dotted*, *accidental*, *tie*, dan bendera sambung.

## PENGUJIAN PERBANDINGAN DENGAN PARTITUR

Pengujian dilakukan dengan membandingkan partitur hasil *output* dari aplikasi ini dengan partitur yang sebenarnya. Partitur lagu "Alabama" yang asli dapat dilihat pada Gambar 14, sedangkan hasil pembacaan file midi lagu tersebut menghasilkan partitur seperti terlihat pada Gambar 15.

Dari Gambar 14 dan Gambar 15 dapat dilihat bahwa aplikasi dapat menggambarkan notasi balok sesuai dengan data yang terdapat pada *file* midi, tetapi masih terdapat perbedaan dengan partitur yang sebenarnya, yaitu perbedaan penempatan not pada bar *treble* dan bar *bass*.



Gambar 14. Partitur Lagu "Alabama" Yang Sebenarnya, *Free sheet music, riffs, lessons and tools for musicians who play.* <http://www.8notes.com>



Gambar 15. Partitur Lagu "Alabama" Hasil *Output* Perangkat Lunak

## KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa:

- Aplikasi menempatkan not pada bar dan ketukan sesuai dengan data waktu yang terdapat dalam *file* midi.
- Meskipun telah dilakukan pengolahan dalam menentukan posisi not pada bar *treble* atau *bass*, namun kesalahan dalam peletakan not pada bar *treble* atau *bass* masih dapat terjadi.
- Aplikasi sudah dapat menentukan nilai dan posisi tanda henti, akan tetapi aplikasi belum dapat dapat menentukan urutan simbol tanda henti yang sesuai dengan ketukan.
- Aplikasi tidak bisa mendeteksi nada dasar dari *file* midi. Keterbatasan ini dikarenakan pada teori *file* midi, *key signature event* dapat bernilai negatif, sedangkan nilai *byte hexadecimal* pada *file* midi tidak dapat diisi dengan nilai negatif.
- Perangkat lunak tidak bisa memberi tanda legato, karena pada *file* midi tidak terdapat data untuk lambang tersebut.

## DAFTAR PUSTAKA

1. *Free sheet music, riffs, lessons and tools for musicians who play.* Maret 19, 2007. [http://www.8notes.com/MIDI\\_file\\_format](http://www.8notes.com/MIDI_file_format). Februari 21, 2007. <http://www.sonicspot.com/guide/midifiles.html>
2. Treblis Software's. *Music notation reference guide.* Maret 25, 2007. <http://www.treblis.com/index.htm>
3. Conger, J., 1988, *C programming for MIDI.* Redwood City, California: M&T Publishing, Inc,
4. *Musical Instrument Digital Interface.* Maret 25, 2007. <http://en.wikipedia.org/wiki/Midi#Overview>
5. White, D., *A crash course on the standard MIDI specification.* Februari 21, 2007. <http://www.skytopia.com/project/articles/midi.html>